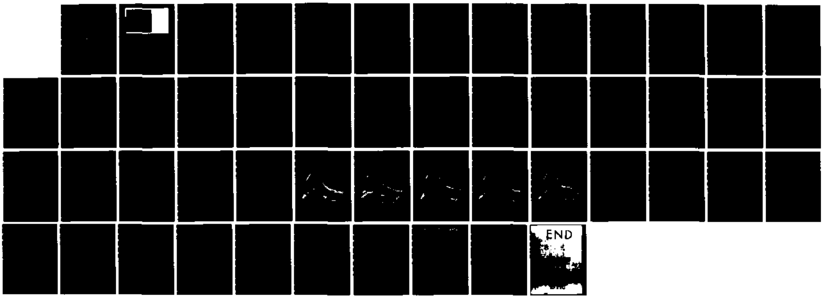
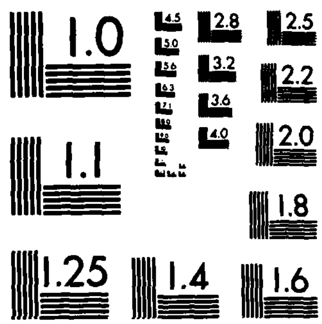
AD-A144 663 MULTIVARIATE SCATTERED DATA DERIVATIVE GENERATION BY FUNCTIONAL MINIMIZATION(U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER P ALFELD JUN 84 1/1

UNCLASSIFIED MRC-TSR-2703 DAAG29-80-C-0041 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MRC Technical Summary Report #2703

MULTIVARIATE SCATTERED DATA DERIVATIVE GENERATION BY FUNCTIONAL MINIMIZATION

Peter Alfeld

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

June 1984

(Received May 22, 1984)

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

Department of Energy
Washington, D. C. 20545

-i-

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

MULTIVARIATE SCATTERED DATA DERIVATIVE GENERATION BY FUNCTIONAL MINIMIZATION

Peter Alfeld*

Technical Summary Report #2703
June 1984

ABSTRACT

Many multivariate interpolation schemes require as data values of derivatives that are not available in a practical application, and that therefore have to be generated suitably. A specific approach to this problem is described that is modeled after univariate spline interpolation. Derivative values are defined by the requirement that a certain functional be minimized over a suitable space subject to interpolation of given positional data. In principle, the technique can be applied in arbitrarily many variables. The theory is described in general, and particular applications are given in one and two variables. A major tool in the implementation of the technique is the Bézier-Bernstein form of a multivariate polynomial. The technique yields visually pleasing surfaces and is therefore suitable for design applications. It is less suitable for the approximation of derivatives of a given function.

AMS (MOS) Subject Classification: 65D05

Key Words: Scattered Data, Computer Aided Geometric Design, Bivariate Interpolation, Derivative Generation

Work Unit Number 3 (Numerical Analysis and Scientific Computing)

---

*Department of Mathematics, University of Utah, Salt Lake City, Utah 84112.

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041, and the Department of Energy under Contract No. DE-AC02-82ER12046 A000.

SIGNIFICANCE AND EXPLANATION

This report addresses the problem of the interpolation of bivariate scattered data (x_i, y_i, z_i) as they arise for example in the Computer Aided Geometric design of the shape of a vehicle or in the representation of measured data. Many techniques proceed by triangulating the domain and then defining the interpolant piecewise on each individual triangle. However, such interpolation schemes usually require as data values of derivatives that the user is unable to supply. The report describes an approach to the generation of such data from the given information. It is modeled after univariate cubic spline interpolation and consists of choosing that interpolant which minimizes a suitable functional (modeling e.g. the strain energy of a clamped elastic plate) over a suitable function space, subject to interpolation. The user has some choice in the selection of the functional. The technique has the advantage of yielding visually pleasant surfaces. It has the drawback that the derivative generation is a global process that depends upon all given data. The scheme can be considered either a method of constructing a once differentiable interpolant to positional data, or a method of constructing derivative values through second order that can then be used to generate a twice differentiable surface by employing another interpolation method. The basic ideas are also illustrated in the univariate context.

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

MULTIVARIATE SCATTERED DATA DERIVATIVE GENERATION BY FUNCTIONAL MINIMIZATION

Peter Alfeld*

1. INTRODUCTION

Many multivariate interpolation schemes require values of derivatives as data which are not available in a practical application. For examples of such schemes, see, e.g., Barnhill and Farin, 1981, the papers by Alfeld and by Alfeld and Barnhill, and other papers in the proceedings edited by Barnhill and Nielson, 1984, and the finite element literature. Thus, if an interpolation problem is to be solved, derivative data must be generated from existing positional data.

In this paper, a particular technique for derivative generation is described which is modeled after univariate spline interpolation. In the univariate case, it is well known that the interpolating spline minimizes a variational principle, and it has been recognized (see Dube 1975, Duchon 1976, and Nielson 1983) that a similar property is desirable also in the multivariate context. Minimizing an appropriate functional over as large a space as possible has the drawback that usually this is difficult computationally and that in general the result will not be a function with a simple and convenient (e.g. polynomial) structure. In our approach, a simple structure is forced by choosing the interplants from a class of piecewise polynomial functions (defined over a tesselation into simplices, i.e. intervals or triangles) such that some user specified functional is minimized. The approach is related to the finite element method for the solution of variational problems, and indeed many computational techniques familiar in that context can also be used for derivative generation. The technique is mathematically well founded and yields visually pleasing surfaces. This makes it suitable for design purposes. Its main drawback is that the derivative generation is a global process that requires the

*Department of Mathematics, University of Utah, Salt Lake City, Utah 84112.

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041, and the Department of Energy under Contract No. DE-AC02-82ER12046 A000.

construction and solution of a large sparse system of linear equations. After the preprocessing stage of derivative generation, however, the schemes are local, i.e. the evaluation of the interpolant requires data only on that simplex in which the data point resides .

The techniques presented here can be viewed in at least two different ways. The point of view that is reflected in the title is that they simply generate derivative values at points. Also possible is the view that the derivative generation and the interpolation are an entity that allows the user to construct smooth interpolants to positional data, without requiring user specified derivative values. From that point of view, the fact that the derivative generation may be divorced from the interpolation, and derivatives may be used for separate purposes (e.g. generating a smoother surface), is a serendipitous byproduct of the technique.

In traditional numerical analysis, one usually thinks of an interpolation scheme as a means of approximating an underlying smooth function. This, however, is often inappropriate in multivariate applications. Rather, the attitude usually taken by designers and engineers is that one is given some data with the objective of passing a visually pleasing surface through the data. Indeed, in a design problem, the primitive function is the object to be created, rather than reproduced or approximated. The designer may even change the positional data in order to get a better surface. This is why the title of this paper contains the term "generation" rather than "estimation". We are interested not in estimating derivatives that have no meaning in the context of the given problem, but rather in trying to determine free parameters in a way that would satisfy a designer or engineer. This attitude is the main reason why the question of errors is not addressed in this paper.

The paper is organized as follows:

In section 2, we develop the theoretical frame-work for the technique. We define the notation and address the questions of invariance of the interpolants under certain geometrical transformations, of existence and uniqueness of the minimizing interpolant, and of the maximum degree of the polynomials that are reproduced exactly by the method. This frame-work applies to functions of arbitrarily many variables.

In section 3, we consider the univariate case. We describe (piecewise cubic and quintic) Hermite interpolation and give formulas for picking the values of the derivatives so as to minimize the norm of certain derivatives of the interpolant. These include natural cubic and quintic splines as special cases.

In section 4, we go on to functions of two variables. This section is the heart of the paper. We consider one basic interpolant (defined over a triangulation), namely a piecewise quintic C^1 interpolant that interpolates to gradients and Hessians (as well as to position). The interpolant has been long known in the finite element context.

In all cases, positional data are considered given, and derivative values are considered parameters to be picked so as to minimize a functional. The functional in all cases is the L_2 norm of some suitably picked set of derivatives.

In section 5, we illustrate the technique with some bivariate examples.

2. THE FRAME-WORK

In this section, we define some notation that will be used in the sequel. We also describe the approach in general, and observe some simple but fundamental properties. The following sections can then be considered specific examples for our general approach. Some of our definitions and concepts could be formulated more generally, but we restrict them to situations as they arise in practical applications.

We denote by D a polyhedral region in $\mathbb{R}^d$ that has been tesselated into simplices. We do not address the question of how this tesselation may be accomplished. See Little, 1983, for some answers to this question.

Our interpolant will be piecewise polynomial (so that integrations can be carried out easily). We denote by B_n^m a space of functions that are polynomial on each of the simplices in D and that are globally m times continuously differentiable. The space contains all polynomials of degree up to and including n. In one variable, B_n^m will not contain any higher degree polynomials, but in more than one variable, B_n^m may contain some higher degree polynomials, depending upon the application.

We assume that our interpolant is required to interpolate to N data d_i defined by linear functionals, thus, if p denotes the interpolant,

$$L_i p = d_i \quad , \quad i = 1,\ldots,N \tag{2.1}$$

In most applications, the L_i will be point evaluations at the vertices of the tesselation. We also assume that the interpolant contains free parameters p_i that we have to specify. In most applications, these parameters will be values of derivatives at points. We express the link between the interpolants and the parameters in terms of linear functionals:

$$(2.2) \qquad P_i p = \rho_i , \quad i = 1,\ldots,M \quad .$$

We collect the data d_i into the vector d and now define the affine space that contains all candidates for the final interpolant by

$$A_n^m(d) = \{q \in B_n^m : \text{(2.1) and (2.2) hold where } \rho_i \in \mathbb{R} \}$$

Fundamental to our technique is the minimization of certain derivatives. We consider the differentiation operator

$$D^k q = \sum_{i_1, i_2, \ldots, i_k \in \{1,\ldots,d\}} \left(\frac{\partial^k q}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_k}} \right)^2 .$$

We also use the functional defined on B_n^m by

$$(2.3) \qquad F^k q = \sum_{\{T\}} \int_T D^k q$$

Here, the summation is over all simplices in the tesselation of D and the integration is over an individual simplex. For example, if $k = 1$, we obtain the Eucledian norm of the gradient, and if $k = 2$, we obtain the Frobenius norm of the Hessian of q.

In the now defined context, of the infinitely many interpolants contained in $A_n^m(d)$, we choose that one that minimizes F^k.

The following theorems cover some of the relevant properties of the above approach.

Theorem 1: There exists a unique minimizer of F^k in $A_n^m(d)$ for all d if and only if the only function in $A_n^m(0)$ piecewise of degree $< k$ is the zero function.

Remark: Thus we reduce the question of existence and uniqueness for the minimization problem to a similar question for a much simpler Lagrange interpolation problem.

Theorem 2: If p minimizes F^k over $A_n^m(d)$ and p is unique, then p is invariant under translations, isotropic rescalings, and rotations.

Theorem 3: If p minimizes F^k over $A_n^m(d)$, and p is unique, $k < n$, and $d_i = L_i q$ for some polynomial q of degree $< k$, then $p = q$.

Proof of Theorem 1: The functional F^k is quadratic and minimizing it requires the solution of a linear system. If $d = 0$, then the linear system is also homogeneous (since the zero function is a solution). Moreover, the coefficient matrix is independent of d (as can be seen by writing the interpolant in cardinal form). A linear system with arbitrary right hand side possesses a unique solution if and only if the corresponding homogeneous linear system possesses a unique solution. If the zero function is the only function in A_n^m piecewise of degree $< k$, then the solution is unique as any interpolant of a higher degree would yield a non-zero value for F^k.

Proof of Theorem 2: We consider a change of variables $z = Lx+b$ where $L^TL = \alpha^2 I$ for some constant α, , with I being the identity matrix. This class of transformations includes translations ($L = 0$), isotropic rescaling ($L = \alpha I$, $b = 0$), and rotations ($\alpha = 1$, $b = 0$) . We have to show that the solution of the minimization problem is independent under this change of variables.

Let $\overline{p}(x) = p(Lx+c)$. We will show that

$$D^k\overline{p} = \alpha^{2k} D^k p \quad . \tag{2.4}$$

Thus the value of the functional $F^k p$ is proportional to $F^k\overline{p}$ which establishes the theorem.

To see (2.4) requires a simple but messy manipulation of summations. $D^k\overline{p}$ is obviously independent of b , so we may assume $b = 0$. We denote the (i,j) entry of L by L_{ij} . We next see by induction that

$$\frac{\partial^k}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_k}} p(Lx + b) = \sum_{j_1,\ldots,j_k} \prod_{r=1}^{k} L_{i_r j_r} \, p_{x_{j_1} x_{j_2} \cdots x_{j_k}} (Lx + b) \ .$$

In the following, subscripts of p denote partial derivatives, and all evaluations take place at $Lx+b$.

Using this relation, we obtain:

$$D^k \bar{p} = \sum_{i_1, i_2, \ldots, i_k} \left(\frac{\partial^k p(Lx+b)}{\partial x_{i_1} \cdots \partial x_{i_k}}\right)^2 =$$

$$= \sum_{\substack{\mu_1 \ldots \mu_k \\ \nu_1 \ldots \nu_k \\ j_1 \ldots j_k}} L_{\mu_1 j_1} \cdots L_{\mu_k j_k} L_{\nu_k j_k} P_{\mu_1 \ldots \mu_k} P_{\nu_1 \ldots \nu_k}$$

$$= \sum_{\substack{\mu_1 \ldots \mu_k \\ \nu_1 \ldots \nu_k}} \sum_{j_1} L_{\mu_1 j_1} \sum_{j_2} L_{\mu_2 j_2} L_{\nu_2 j_2} \cdots$$

$$\sum_{j_k} L_{\mu_k j_k} L_{\nu_k j_k} P_{\mu_1 \ldots \mu_k} P_{\nu_1 \ldots \nu_k} = \alpha^{2k} \sum_{\mu_1 \ldots \mu_k} P^2_{\mu_1 \ldots \mu_k} = \alpha^{2k} D^k p$$

Proof of Theorem 3: Suppose that the data d_k are generated by a polynomial q of degree $< k < n$. Then $q \in A_n^m(d)$ and ${}^k q$ assumes the minimum possible value zero. Since the minimizer is unique we have that $p = q$.

In the sequel, we will make use of a technique familiar from the finite element technique (see, e.g., Strang and Fix, 1973, p.28). Rather than integrating over the entire domain D and then differentiating with respect to the parameters, we integrate over an individual simplex (element), differentiate with respect to those parameters that contribute nontrivially on the element, and in this fashion obtain an element stiffness system. The element stiffness systems then have to be assembled into the global stiffness system, and that system has to be solved. We assume the assembly technique to be familiar. Also, we will not address the question of how the (large, sparse, positive definite) global stiffness system may be solved except to point out that this is usually done by direct methods and that sophisticated software for this purpose exists (see George and Liu, 1981).

3. UNIVARIATE APPLICATIONS

3.1 The C^1 Case

Suppose we are given data (x_n,f_n) , $n = 1,\ldots,N$, where

$$a = x_1 < x_2 < \ldots < x_N = b$$

We denote the interval $[x_n,x_{n+1}]$ by I_n and write $h_n = x_{n+1}-x_n$ (for $n = 1,2,\ldots,N-1$) . Our objective is to interpolate to the given data, i.e. we are looking for a function s , say, that satisfies

$$s(x_n) = f_n, \quad \text{for all } n = 1,2,\ldots,N \tag{3.1}$$

In this subsection, we require s to be in the space B_3^1 , , i.e. the space of all functions that can be represented as a cubic polynomial on each subinterval I_n , and that are once continuously differentiable on $I: = [a,b]$. In the notation of section 2, we let $d_i=f_i$ and $p_i = p'(x_i)$ where p is the interpolant. The affine space A_3^1 is the space of all functions in B_3^1 , that satisfies the interpolation condition (3.1). The functional F^k (for $k = 0,1,2,3$) becomes

$$F^k p = \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} (p^{(k)})^2 \, dx \tag{3.2}$$

To make the notation of the following results more compact, we consider the following more general functional

$$G^k p = \sum_{k=0}^{3} w_k F^k p$$

where the w_k are weight factors that can be chosen by the user. For instance, the choice $w_2 = 1$, $w_0 = w_1 = w_3 = 0$ yields the natural interpolatory cubic spline (which, remarkably, is twice continuously differentiable everywhere on I).

The differentiation and integration on an individual subinterval was carried out using the symbol manipulation language REDUCE (Hearn, 1983). We obtain the element stiffness matrix

$$A_n = c \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \quad \text{and the right hand side} \quad g_n = c \begin{vmatrix} r_1 \\ r_2 \end{vmatrix}$$

where

$$c = \frac{1}{210\, h_n^3}$$

$$a_{11} = a_{22} = 4h_n^6 w_0 + 56h_n^4 w_1 + 1680h_n^2 w_2 + 15120\ w_3$$

$$a_{12} = a_{21} = -3h_n^6 w_0 - 14h_n^4 w_1 + 840h_n^2 w_2 + 15120\ w_3$$

(3.3)

$$r_1 = \frac{1}{h} [-h_n^6(22f_{n+1} + 13f_n)w_0 + 42h_n^4(f_{n+1}-f_n)w_1$$

$$+ 2520h_n^2(f_{n+1}-f_n)w_2 + 30240(f_{n+1}-f_n)w_3]$$

$$r_2 = \frac{1}{h} [+h_n^6(22f_{n+1} + 13f_n)w_0 + 42h_n^4(f_{n+1}-f_n)w_1$$

$$+ 2520h_n^2(f_{n+1}-f_n)w_2 + 30240(f_{n+1}-f_n)w_3]$$

The formulas given in (3.2) and (3.3) can be assembled into a global stiffness system whose solution defines a piecewise cubic Hermite interpolant that minimizes the weighted derivatives defined in (3.2).

Of special interest is the case where $w_3 = 1$ and the other weights equal zero. The element stiffness matrix is of rank one. The assembly procedure yields a consistent system of equations with rank deficiency one. Thus there is a one parameter family of minimizing interpolants. That this is so can also be recognized directly: Any piecewise quadratic interpolant would minimize the norm of the third derivative. A quadratic polynomial has three degrees of freedom. On the first subinterval, we have to interpolate to two function values, retaining one free parameter. On each subsequent interval, we have to interpolate to two function values, and also have to match the derivative of the quadratic on the left neighboring interval, using up all three parameters. Thus, overall one parameter remains at our disposal. The significance of this observation is that quadratic precision of a cubic Hermite scheme cannot be achieved by specifying the functional (3.2) alone. Rather, an additional condition has to be imposed to eliminate the free parameter. A simple approach would consist of determining the first derivative at some grid point by using a quadratically precise interpolation formula, and then requiring the minimizing interpolant to reproduce that derivative.

3.2 The C^2 Case

The development in this subsection parallels that in the preceding one, except that we consider interpolants in B_5^2 , and also consider second derivatives of the interpolant at the gridpoints as parameters. The parameters at our disposal are $d_n = p'(x_n)$ and $d_{n+N} = p''(x_n)$ for $n = 1,\ldots,N$. The affine space A_5^2 is the space of all functions in B_5^2 that satisfy the interpolation condition (3.1).

Similarly as before, we consider the general functional

$$\mathcal{H}^k p = \sum_{k=0}^{5} w_k \mathcal{F}^k p$$

where the $\mathcal{F}^k$ are defined in (3.2). Carrying out the integration and differentiation on a subinterval I_n and writing $h = h_n$,

$$df = f_{n+1} - f_n \text{ , } p_n' = p'(x_n) \text{ , } p_n'' = p''(x_n)$$

yields the element stiffness equation:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{vmatrix} \begin{vmatrix} p'_n \\ p''_n \\ p'_{n+1} \\ p''_{n+1} \end{vmatrix} = \begin{vmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{vmatrix}$$

where

$$a_{11} = (832c_0 + 288c_1 + 384c_2 + 64c_3 + 456c_4 + 216c_5)h$$

$$a_{12} = (69c_0 + 21c_1 + 22c_2 + 12c_3 + 78c_4 + 36c_5)h^2$$

$$a_{13} = (-532c_0 - 18c_1 + 216c_2 + 56c_3 + 444c_4 + 216c_5)h$$

$$a_{14} = (52c_0 + 6c_1 - 8c_2 - 8c_3 - 72c_4 - 36c_5)h^2$$

$$a_{22} = (6c_0 + 2c_1 + 6c_2 + 3c_3 + 14c_4 + 6c_5)h^3$$

$$a_{23} = (-52c_0 - 6c_1 + 8c_2 + 8c_3 + 72c_4 + 36c_5)h^2$$

$$a_{24} = (5c_0 + c_1 + c_2 - c_3 - 11c_4 - 6c_5)h^3$$

$$a_{33} = (832c_0 + 288c_1 + 384c_2 + 64c_3 + 456c_4 + 216c_5)h$$

$$a_{34} = (-69c_0 - 21c_1 - 22c_2 - 12c_3 - 78c_4 - 36c_5)h^2$$

$$a_{44} = (6c_0 + 2c_1 + 6c_2 + 3c_3 + 14c_4 + 6c_5)h^3$$

$$g_1 = (-1812f_{n+1} - 3732f_n)c_0 + (270c_1 + 600c_2 + 120c_3 + 900c_4 + 432c_5)(f_{n+1} - f_n)$$

$$g_2 = h[(-181f_{n+1} - 281f_n)c_0 + (15hc_1 + 30hc_2 + 20hc_3 + 150hc_4 + 72hc_5)(f_{n+1} - f_n)]$$

$$g_3 = (3732f_{n+1} + 1812f_n)c_0 + (270c_1 + 600c_1 + 120c_3 + 900c_4 + 432c_5)(f_{n+1} - f_n)$$

$$g_4 = h[(-281f_{n+1} - 181f_n)c_0 - (15hc_1 + 30hc_2 + 20hc_3 + 150hc_4 + 72hc_5)(f_{n+1} - f_n)]$$

$$c_0 = \frac{h^2}{27720}w_0,\ c_1 = \frac{1}{630}w_1,\ c = \frac{h^{-2}}{35}w_2,\ c_3 = 6h^{-4}w_3,\ c_4 = 48h^{-6}w_4,\ c_5 = 1200h^{-8}w_5.$$

The following cases are of special interest:

$w_i = \delta_{i2}$: The element stiffness matrix is non-singular. The unique solution for $N = 1$ is the linear interpolant. In general, the interpolant minimizes the norm of the second derivative over the space B_5^2 which contains the interpolating natural cubic spline. Since the spline minimizes the norm of the second derivative over the larger space containing all piecewise C^2 functions that are once differentiable at the grid points, the piecewise quintic Hermite interpolant generated here actually is the interpolating natural cubic spline.

$w_i = \delta_{3i}$: The element stiffness matrix is of rank 3. In the case of $N = 2$ any interpolating quadratic will minimize the functional H^k. In the case $N = 3$, the interpolant is the unique interpolating quadratic which renders C^2 zero. The global stiffness matrix is non-singular whenever $N > 2$. In that case, the interpolant is the natural quintic spline, which, remarkably, is four times continuously differentiable everywhere in I.

$w_i = \delta_{4i}$: The global stiffness matrix is rank deficiency 2 for any value $N > 1$. Any piecewise cubic interpolant will render H^k zero.

$w_i = \delta_{5i}$: The global stiffness matrix is of rank deficiency $N+3$ for any $N > 1$. Any piecewise quartic interpolant will render the value of H^k zero.

4. BIVARIATE APPLICATIONS

4.1 Barycentric Coordinates and the Bézier form of a Polynomial

Conceptually, we proceed as in the univariate case. However, the algebraic manipulations grow much more complicated. The problem becomes tractable only by using barycentric coordinates and the Bézier form of a polynomial. We develop these concepts for the bivariate case. The generalization to more than two variables is straightforward.

We restrict our attention to a general triangle T with vertices V_1, V_2, and V_3, and express the location of a point P in T by

$$P = \sum_{i=1}^{3} b_i V_i \quad \text{where} \quad \sum_{i=1}^{3} b_i = 1 \tag{4.1}$$

The b_i are the barycentric coordinates of P. They are determined uniquely by the linear system (4.1) provided the underlying triangle T is non-degenerate (which we will assume throughout).

We write a polynomial p of degree n in the Bezier form

$$p = \sum_{r+s+t=n} \frac{n!}{r!s!t!} c_{rst}\, b_1^r b_2^s b_3^t \tag{4.2}$$

The computation of the element stiffness system requires the following two ingredients:

1. Spatial differentiation of the Bézier form of a polynomial.

2. Integration of the Bézier form over a general triangle.

We address each ingredient in turn:

4.1.1 Spatial Differentiation of the Bézier form of a Polynomial.

We let $\mathcal{D}$ denote a spatial differentiation operator, i.e. $\mathcal{D} = \frac{\partial}{\partial e}$ for some direction e. We consider the general Bézier form given by (4.2). Then $\mathcal{D}p$ is a polynomial of degree $n-1$ which can be written in the form

$$\mathcal{D}p = \sum_{r+s+t=n-1} \frac{(n-1)!}{r!s!t!} \bar{c}_{rst} \, b_1^r b_2^s b_3^t$$

It can be easily verified (see Farin, 1980) that

$$\bar{c}_{rst} = n[c_{r+1,s,t} \mathcal{D}b_1 + c_{r,s+1,t} \mathcal{D}b_2 + c_{r,s,t+1} \mathcal{D}b_3]$$

In our application, we are interested in differentiating p several times in different directions. For later reference we list the relevant formulas. Let $\mathcal{D}_i$, $i = 1,\ldots,k$, denote spatial directional derivatives (such as the partial derivatives with respect to x and y). Then

$$\mathcal{D}_k \mathcal{D}_{k-1} \cdots \mathcal{D}_1 p = \sum_{r+s+t=n-k} \frac{(n-k)!}{r!s!t!} c^{(k)}_{rst} b_1 b_2 b_3$$

where

$$c^{(0)}_{rst} = c_{rst} \tag{4.3}$$

and

$$c^{(i+1)}_{rst} = (n-i)[c^{(i)}_{r+1,s,t}\mathcal{D}_{i+1} b_1 + c^{(i)}_{r,s+1,t}\mathcal{D}_{i+1} b_2 + c^{(i)}_{r,s,t+1}\mathcal{D}_{i+1}\ b_3] \tag{4.4}$$

for $i = 0,1,\ldots,k$.

4.1.2 Integration of the Bézier form over a General Triangle

Let p be defined by (4.2). The polynomial is defined in terms of barycentric coordinates, but integration is required in terms of cartesian coordinates. A simple change of variables (see, e.g., Lang, 1968, p.421) yields for any integrable function f

$$\iint_T f(P)\,dxdy = a(T)\int_0^1 \int_0^{1-b_1} f(b_1V_1 + b_2V_2 + (1-b_1-b_2)V_3)\,db_2\,db_1$$

where T is the triangle of interest and $a(T)$ is twice the area of T.

Now let p denote a general Bézier representation given by (4.2). Using the substitution $b_3 = 1-b_1-b_2$, it is easy to see by induction and integration by parts that

$$\int_0^1 \int_0^{1-b_1} p\,db_2\,db_1 = \frac{1}{(n+1)(n+2)} \sum_{r+s+t=n} c_{rst}$$

4.2 A Q_{18} based Interpolant

The underlying interpolant here is the Bézier form of Q_{18}, a reduced quintic interpolant requiring function values, gradients, and Hessians at the vertices of T. The overall piecewise quintic interpolant is continuously differentiable everywhere in the triangulation. For the derivation of an explicit expression for Q_{18} see Barnhill and Farin, 1981. For completeness and reference we list the relevant formulas, also introducing the notation used in the sequel. The interpolant is given by:

$$Q = \sum_{r+s+t=5} \frac{5!}{r!s!t!} d_{rst} b_1^r b_2^s b_3^t \tag{4.5}$$

where:

(4.6)

$$d_{500} = F_1, \; d_{050} = F_2, \; d_{005} = F_3$$

$$d_{410} = \frac{1}{5} F_{3,1} + d_{500}, \; d_{401} = \frac{-1}{5} F_{2,1} + d_{500}, \; d_{140} = \frac{-1}{5} F_{3,2} + d_{050}$$

(4.7)

$$d_{041} = \frac{1}{5} F_{1,2} + d_{050}, \; d_{014} = \frac{-1}{5} F_{1,3} + d_{005}, \; d_{104} = \frac{1}{5} F_{2,3} + d_{005}$$

$$d_{320} = \frac{1}{20} F_{33,1} + 2d_{410} - d_{500}, \; d_{230} + \frac{1}{20} F_{33,2} + 2d_{140} - d_{050}$$

$$d_{032} = \frac{1}{20} F_{11,2} + 2d_{041} - d_{050}, \; d_{023} = \frac{1}{20} F_{11,3} + 2d_{014} - d_{005}$$

(4.8)

$$d_{203} = \frac{1}{20} F_{22,3} + 2d_{104} - d_{005} \text{ , } d_{302} = \frac{1}{20} F_{22,1} + 2d_{401} - d_{500}$$

$$d_{311} = \frac{-1}{20} F_{23,1} + d_{410} + d_{401} - d_{500}$$

(4.9)

$$d_{131} = \frac{-1}{20} F_{13,2} + d_{140} + d_{041} - d_{050}$$

$$d_{113} = \frac{-1}{20} F_{12,3} + d_{014} + d_{104} - d_{005}$$

$$d_{122} = \frac{1}{60} [60s_1F_2 - 6F_{3,2} + 6(5s_1 - 1)F_{1,2} - 2F_{13,2}$$

$$+ (5s_1 - 2)F_{11,2} - 60(2_1 - 1)F_3 + 6(5s_1 - 4)F_{1,3}$$

$$+ 6F_{2,3} - (5s_1 - 3)F_{11,3} - 2F_{12,3}]$$

(4.10)

$$d_{212} = \frac{1}{60} [60s_2F_3 - 6F_{1,3} + 6(5s_2 - 1)F_{2,3} - 2F_{12,3}$$

$$+ (5s_2 - 2)F_{22,3} - 60(s_2 - 1)F_1 + 6(5s_2 - 4)F_{2,1}$$

$$+ 6F_{3,1} - (5s_2 - 3)F_{22,1} - 2F_{23,1}]$$

$$d_{221} = \frac{1}{60} [60s_3F_1 - 6F_{2,1} + 6(5s_3 - 1)F_{3,1} - 2F_{23,1}$$

$$+ (5s_3 - 2)F_{33,1} - 60(s_3 - 1)F_2 + 6(5s_3 - 4)F_{3,2}$$

$$+ 6F_{1,2} - (5s_3 - 3)F_{33,2} - 2F_{13,2}]$$

and:

(4.11)

$$F_i = Q(V_i)$$

(4.12)

$$F_{k,i} = \frac{\partial Q}{e_k}(V_i)$$

(4.13)

$$F_{jk,i} = \frac{\partial^2 Q}{e_j e_k}(V_i)$$

(4.14)

$$s_i = -\frac{e_{i+1}^T e_i}{e_i^T e_i}$$

and $e = V_{i+1} - V_i$ (where index arithmetic is modulo 3).

Thus, the data in (4.11) are given positional values, the parameters in (4.12) and (4.13) are derivatives (of the global interpolant) at vertices which are to be determined so as to minimize a suitable functional defined below, and the parameters defined in (4.14) are projections of edges onto other edges that ensure global differentiability of the interpolant.

It is important to keep in mind that ultimately we wish to generate derivative data (i.e. gradients and Hessians) at all vertices of a given triangulation. The local interpolant Q is most conveniently expressed in terms of derivatives in the direction of edges (which would be inappropriate on a neighboring triangle utilizing common vertex data). Hence we need to express edge derivatives in terms of partial derivatives in x and y. This is accomplished by writing:

$$g_i = \nabla Q(V_i),\; H_i = \nabla^2 Q(V_i) \tag{4.15}$$

and

$$F_{k,i} = e_k^T g_i,\quad F_{jk,i} = e_j^T H_i e_k \tag{4.16}$$

We now proceed as in section 3. We consider the space B_4^1 of all functions that can be expressed as Q_{18} on each triangle, and that are globally once continuously differentiable. (Notice that this set contains all quartic polynomials, but only some quintic polynomials). The affine space A_4^1 of interest is the set of all functions s in B_4^1 that satisfy the interpolation requirement

$$s(x_i, y_i) = f_i \;\; (= F_j,\ j \in \{1,2,3\} \text{ on a particular triangle}) \tag{4.17}$$

We want to minimize the functional F^k defined in (2.3) over B_4^1. The parameters at our disposal are the values of gradients and Hessians at the gridpoints (x_i, y_i).

In addition to the items covered in general in section 4.1, we need to differentiate the interpolant with respect to the parameters. As before, we consider a single element, i.e. a single triangle. On each triangle, we have 15 parameters at our disposal. We denote by $v = [v_i]$ the vector of these parameters. Because of the square under the integral in (2.3), the contribution $\phi(v)$, say to F^k from a single triangle is a quadratic function of v. This implies that we can write

$$\phi(v) = \frac{1}{2} v^T A v + g^T v + c$$

where A is a symmetric positive semidefinite 15 by 15 matrix with entries a_{ij}, g is a vector with entries g_i, and c is a constant. The contribution of $\phi(v)$ to the global stiffness system is determined by A and g. Since A is the Hessian of $\phi(v)$ we can write

$$a_{ij} = \frac{\partial^2 \phi}{\partial v_i \partial v_j} \tag{4.19}$$

Once the a_{ij} have been determined, the g can be computed by substituting the unit vectors for v, thus

$$g_i = \phi(e_i) - \frac{1}{2} a_{ii} - c$$

(where e_i is the unit vector with 1 in the i-th place and zeros elsewhere, and $c = \phi(0)$). It is therefore crucial to be able to carry out the differentiation (4.19). Applying the product rule twice yields

$$a_{ij} = 2\iint_T \left(\frac{\partial}{\partial v_i} Q \frac{\partial}{\partial v_j} Q\right) dxdy$$

The differentiation of Q with respect to the entries of v can be carried out by differentiating in the recursion formulas (4.3) and (4.4). Differentiating in (4.4) yields

$$\begin{aligned}\frac{\partial}{\partial v_i} d^{(i+1)}_{rst} = (5-i)\Big[&\frac{\partial}{\partial v_i} d^{(i)}_{r+1,s,t}\, D_{i+1}\, b_1 \\ &+ \frac{\partial}{\partial v_i} d^{(i)}_{r,s+1,t}\, D_{i+1} b_2 \\ &+ \frac{\partial}{\partial v_i} d^{(i)}_{r,s,t+1}\, D_{i+1} b_3\Big]\end{aligned}$$

The recursion is initialized by differentiating in (4.3). The derivatives can be derived easily from the recursions (4.6) through (4.16). However, the situation is complicated by the fact that the interpolant is expressed in terms of barycentric coordinates, whereas the derivatives entering as data are cartesian derivatives. Computational efficiency can be gained by observing that most of the derivatives obtained from (4.4) are zero, and by avoiding having the program multiply by the zero derivatives.

For convenience, the non-zero derivatives are listed in the appendix. They were computed using the symbol manipulation language REDUCE (see Hearn, 1973). In order to avoid typing and type-setting errors, the REDUCE output has been reproduced photographically.

5. BIVARIATE EXAMPLES

In this section, we illustrate the technique described in the preceding sections by a simple bivariate example. In a typical application, no underlying primitive function will be available. However, for the sake of illustration we use a primitive first introduced by Franke, 1982. The surface exhibits several phenomena (two maxima, one minimum, and one saddle point) that may occur in surface defined by a true data set. The primitive function is given by

$$f(x,y) = \frac{3}{4} e^{-((9x-2)^2 + (9y-2)^2)/4}$$

$$+ \frac{3}{4} e^{-((9x+1)^2/49 + (9y+1)/10)}$$

$$+ \frac{1}{2} e^{-((9x-7)^2 + (9y-3)^2)/4}$$

$$- \frac{1}{5} e^{-((9x-4)^2 + (9y-7)^2)}$$

Approximations of f on the unit square were constructed from positional values at only 36 points. Figure 1 shows a contour plot of f together with the data points and the triangulation of the data. The triangulation was constructed using Little's method described in Barnhill, 1977.

Figure 2 shows a hidden line plot of f . All subsequent plots are drawn as viewed from the same point and with the same vertical scale. Figures 3 through 6 show the interpolants obtained by minimizing the first through fourth derivatives respectively. Minimizing second or third derivatives yields a very pleasing surface. The interpolant is somewhat shallower than the primitive function. This is due to the fact that none of the data points is placed directly at the extrema, (cf. Figure 1.) which are therefore poorly represented by the interpolants.

Minimizing the gradient yields a surface that is unlikely to be useful for design applications. However, it remarkably resembles a scene one might encounter in a physical mountain range. It is intriguing to speculate that minimizing the gradient may sometimes be appropriate for generating topographical maps since erosion effects are the harsher the steeper the terrain. Minimizing fourth derivatives generates large undulations and is not a good general purpose method.

Primitive Function - Contours and Triangulation

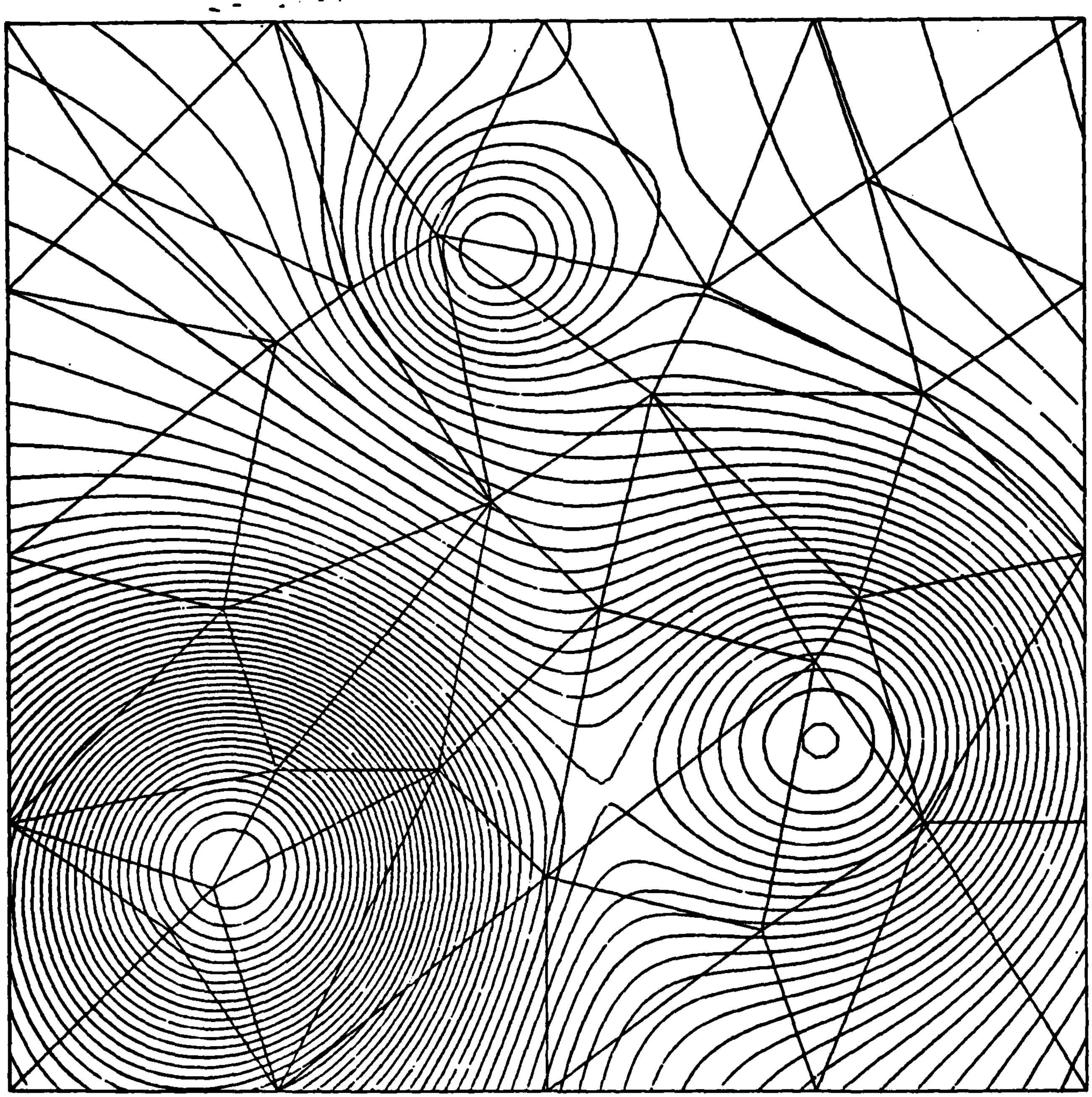


FIGURE 1

Primitive Function

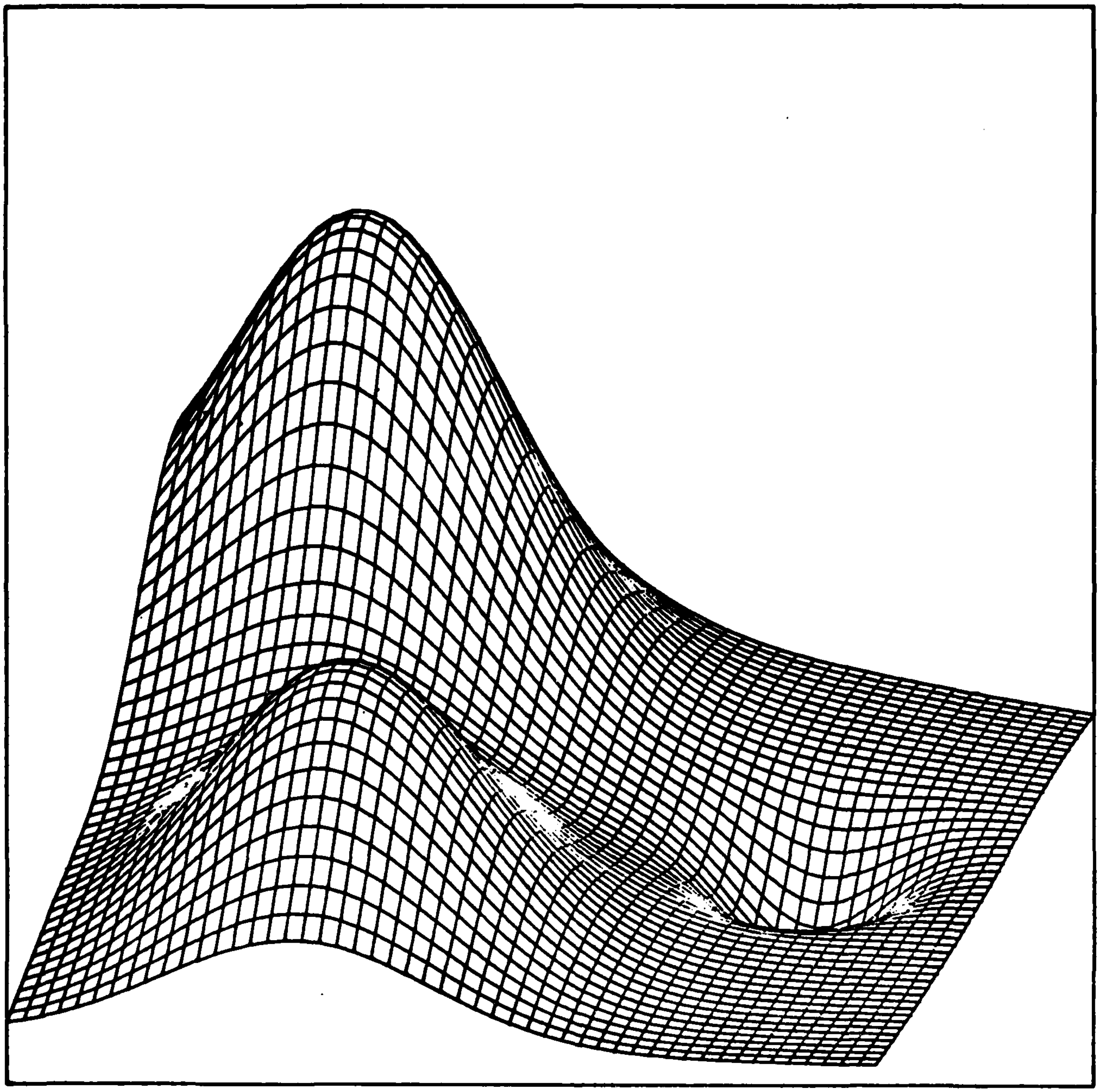


FIGURE 2

Minimization of 1st derivatives

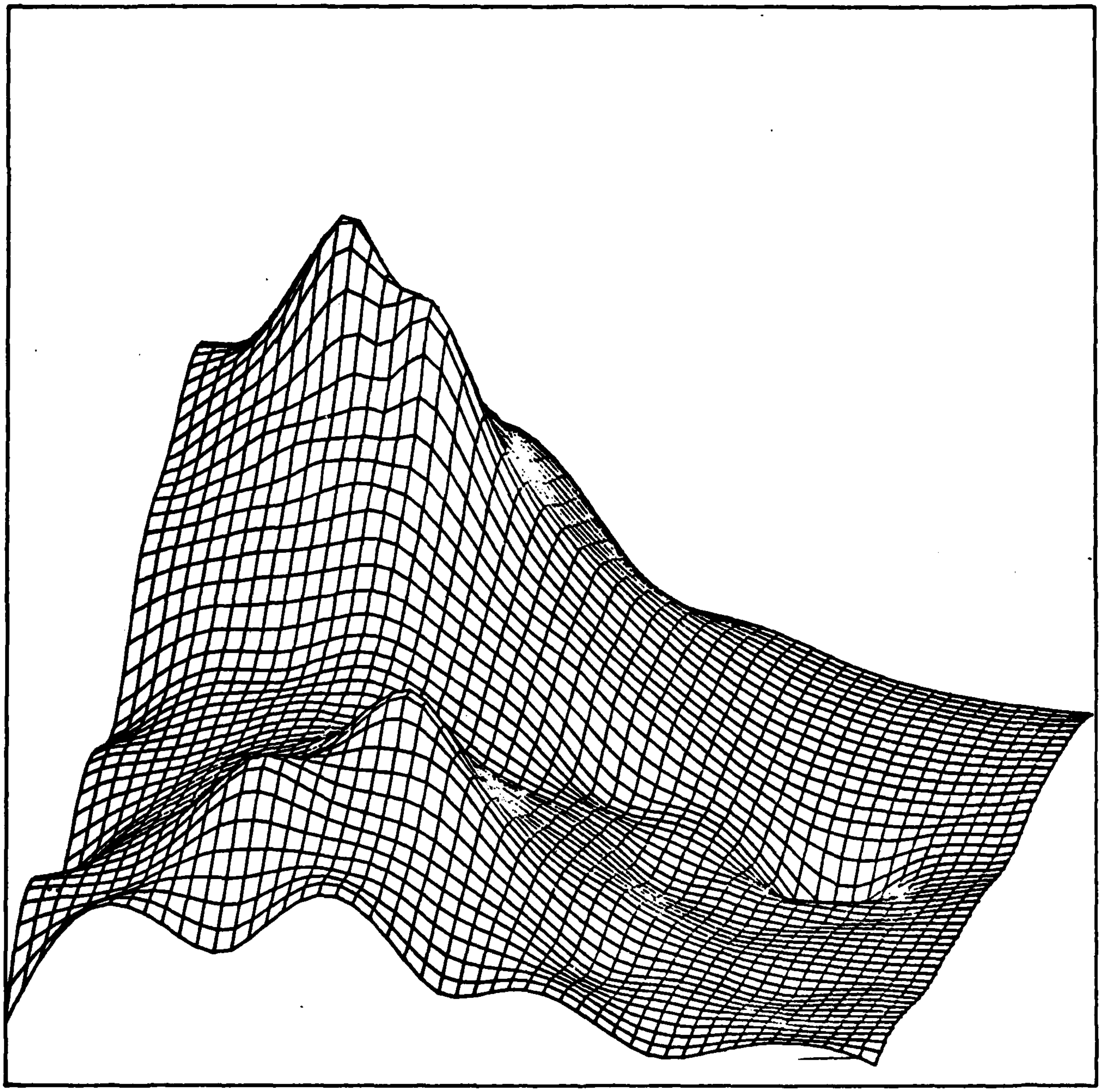


FIGURE 3

Minimization of 2nd derivatives

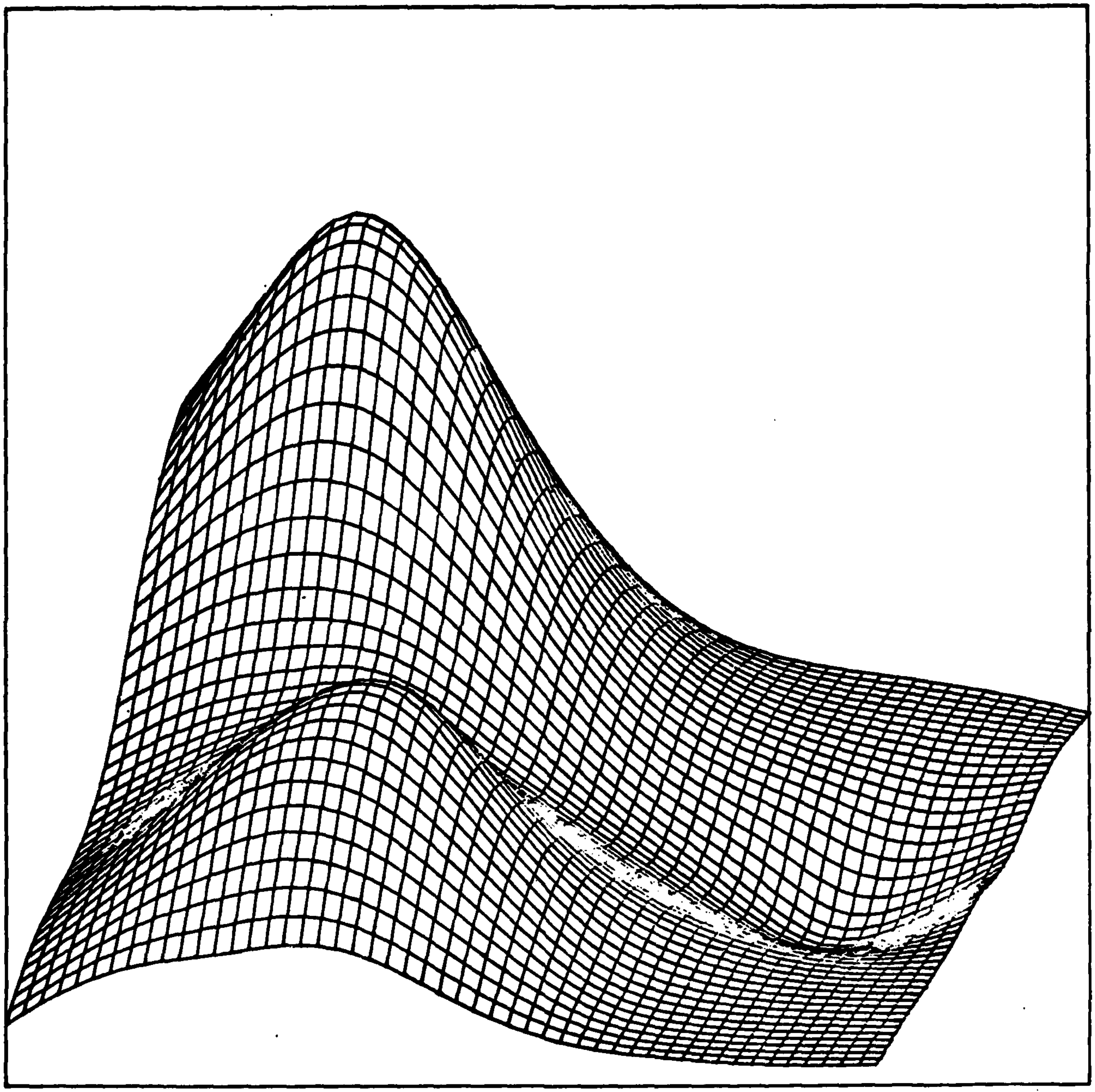


FIGURE 4

Minimization of 3rd derivatives

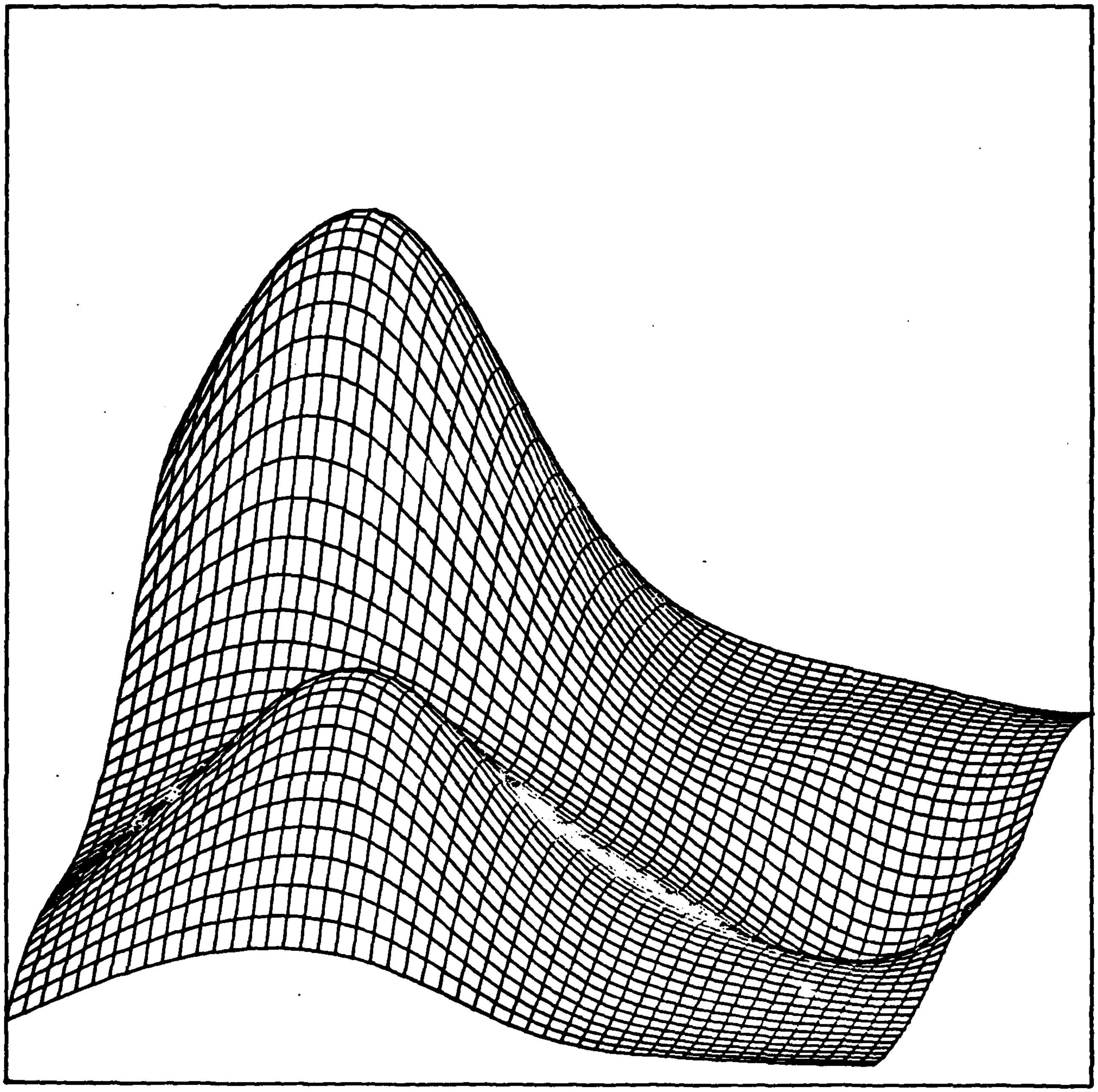


FIGURE 5

Minimization of 4th derivatives

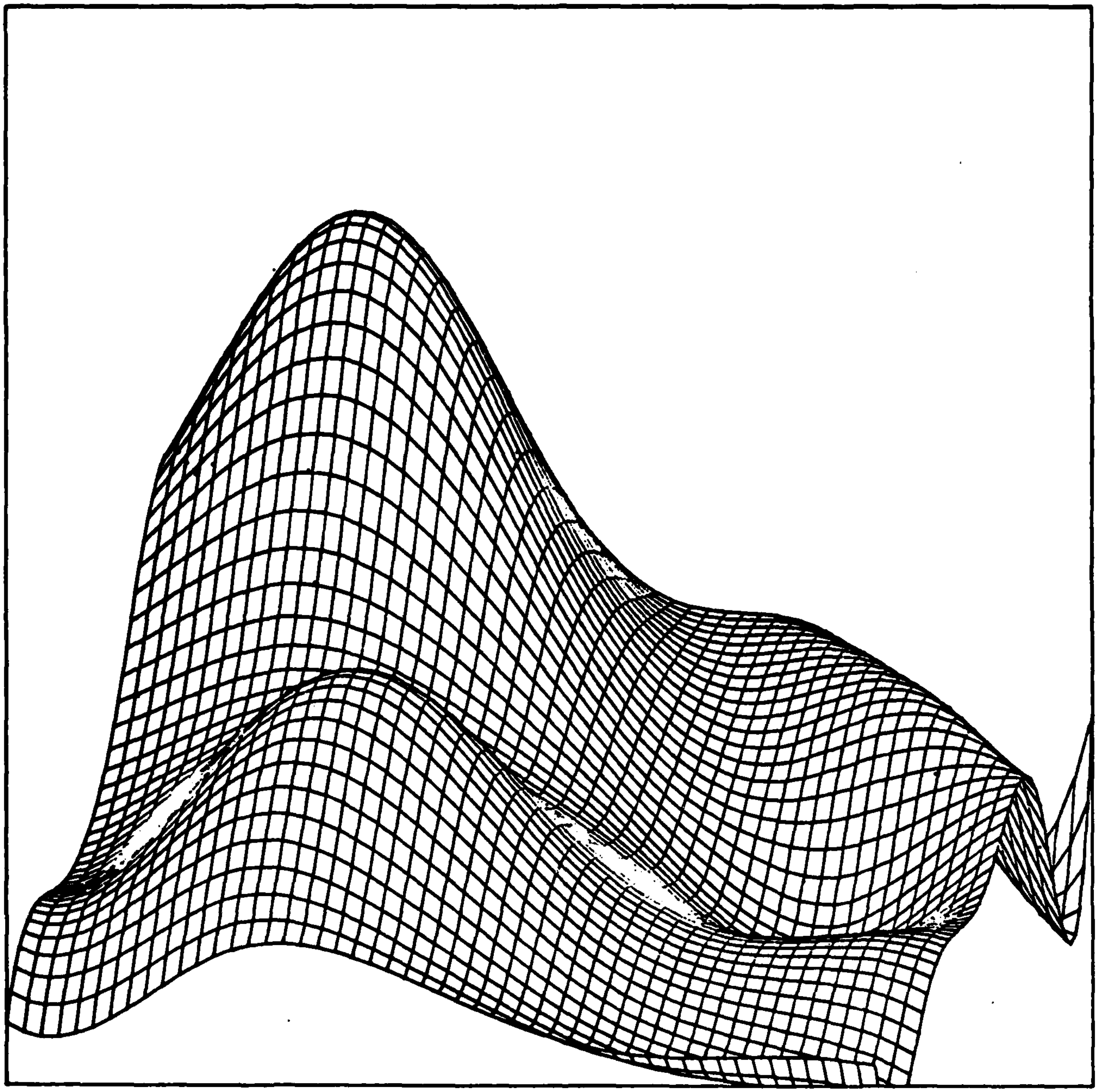


FIGURE 6

CONCLUSIONS

A technique for derivative generation has bee proposed. It is particularly useful in a multivariate context. Its main properties are:

1. The preprocessing stage of derivative generation requires the construction and solution of a large linear system. However, this is closely related to the finite element method, and existing well developed computational techniques can be employed.

2. Once derivatives have been generated, the interpolant is local.

3. The formulas presented here apply to the univariate and the bivariate case only. However, conceptually the extension of the technique to other basic interpolants is straightforward. The particular interpolants considered in this paper are piecewise polynomial functions. This is convenient since the technique calls for the integration of functions with a similar structure as the interpolant. In principle, interpolants with a non-polynomial structure can also be used. They would necessitate either the development of special integration formulas or the use of numerical (multivariate) integration.

4. Primarily, our technique is intended to be used as a means of constructing visually pleasing surfaces. The special application described in section 3. can be considered a bivariate C^1 interpolation scheme that requires only C^0 data. This is remarkable in view of the fact that most existing local multivariate interpolation scheme require data of at least the same degree of smoothness as the degree of smoothness of the interpolant. However, the scheme also supplies values of derivatives at certain points. These

derivative values can be used for other purposes. For example, the gradients and Hessians generated by minimizing our piecewise quintic C^1 interpolant can be used as data for the second of the two piecewise rational C^2 interpolants described in Alfeld, 1984a. That interpolant would of course not possess the minimization property of the polynomial interpolant.

ACKNOWLEDGMENTS

This research was supported by the Department of Energy under contract DE-AC02-82ER12046. A000 to the University of Utah, the United States Army under contract No. DAAG29-80-C0041, a sabbatical leave from the University of Utah, and partly by a University of Utah Research Committee Faculty Grant. The author has benefited from the stimulating environments provided by the Computer Aided Geometric Design Group at the University of Utah headed by R. E. Barnhill and by the Mathematics Research Center of the University of Wisconsin-Madison. A large part of the technical manipulations were carried out using the symbol manipulation language REDUCE (Hearn, 1983). The Figures were generated using the graphics packages PLOT79 (Beebe, 1980). The College of Science of the University of Utah provided computer time.

REFERENCES

Peter Alfeld and R.E. Barnhill (1984), A Transfinite C^2 Interpolant over Triangles, Barnhill and Nielson, 1984, pp. 17-40.

Peter Alfeld (1984a), Two Discrete C^2 Interpolants, appendix to preceding reference.

Peter Alfeld (1984b), A Discrete C^1 Tetrahedral Interpolant, Barnhill and Nielson, 1984, pp. 5-16.

Peter Alfeld (1984c), Multivariate Perpendicular Interpolation, to appear in SIAM Journal on Numerical Analysis.

R. E. Barnhill (1977), Representation and Approximation of Surfaces, in Mathematical Software III, J. R. Rice, ed., Academic Press, pp. 68-119.

R. E. Barnhill and G. Farin (1981), C^1 Quintic Interpolation over Triangles: Two Explicit Representations, Int. J. for Num. Meth. in Eng., v. 17 (1981), pp. 1763-1778.

R. E. Barnhill and G. M. Nielson (ed.), special issue on surfaces, Rocky Mountain Journal of Mathematics, Winter 1984, v. 14, No. 1.

N. H. F. Beebe (1980), A User's Guide to PLOT 79, Departments of Physics and Chemistry, University of Utah, Salt Lake City, UT 84112.

C. de Boor (1978) , A Practical Guide to Splines, Springer Verlag.

R.P. Dube (1975), Local Schemes for Computer Aided Geometric Design, Ph.D. Thesis, Department of Mathematics, University of Utah.

J. Duchon (1976), Fonctions-Spline du type "Plaque Mince" en Dimension 2, Rpt. 27, Univ. of Grenoble.

G. Farin (1980), Bezier polynomials over triangles and the construction of piecewise C^r polynomials, report TR/91, Department of Mathematics, Uxbridge, Middlesex, U.K., 1980.

R. Franke (1982), Scattered Data Interpolation: Test of Some Methods, Math. Comp. 38, pp. 181-200.

A. George and J. W. Liu (1981), Computer Solution of Large Sparse Positive Definite Systems, Prentice Hall.

A.C. Hearn (1973), REDUCE User's manual, 2nd ed., Report UCP-19, Department of Computer Science, University of Utah.

S. Lang (1968), Analysis I, Addison Wesley.

F. F. Little (1983), Triangular Surfaces, Ph.D. Thesis, Department of Mathematics, University of Utah.

G.M. Nielson (1983), A Method for Interpolating Scattered Data Based upon a Minimum Norm Network, Math. Comp. 40, pp. 253-272.

G. Strang and G. Fix (1973), Analysis of the Finite Element Method Prentice Hall, New Jersey, 1973.

A. Zenisek (1973), Polynomial Approximations on Tetrahedrons in the Finite Element Method, Journal of Approximation Theory 7, pp. 334-351.

APPENDIX

Derivatives of the coefficients of Q with respect to the derivative parameters

The following list is computer generated which accounts for the awkward notation:

$$Q = [Qx,Qy]^T, \quad Q^2 = \begin{matrix} Qxx & Qxy \\ Qxy & Qyy \end{matrix},$$

$$s_1 = s1,\ s_2 = s2,\ s_3 = s3$$

$$e_1 = [e1x,e1y]^T,\ e_2 = [e2x,e2y]^T,\ e_3 = [e3x,e3y]^T$$

$$d_{211} = d211,\ d_{121} = d121,\ d_{112} = d112$$

Only non-zero derivatives are listed.

differentiation with respect to Qx(v1):

d212: (5*e2x*s2 - 4*e2x + e3x)/10

d221: (- e2x + 5*e3x*s3 - e3x)/10

d302: (- 2*e2x)/5

d311: (- e2x + e3x)/5

d320: (2*e3x)/5

d401: (- e2x)/5

d410: e3x/5

differentiation with respect to Qy(vl):

```
d212:   (5*e2y*s2 - 4*e2y + e3y)/10

d221:   ( - e2y + 5*e3y*s3 - e3y)/10

d302:   ( - 2*e2y)/5

d311:   ( - e2y + e3y)/5

d320:   (2*e3y)/5

d401:   ( - e2y)/5

d410:   e3y/5
```

differentiation with respect to Qxx(vl):

```
d212:   (e2x*( - 5*e2x*s2 + 3*e2x - 2*e3x))/60

d221:   (e3x*( - 2*e2x + 5*e3x*s3 - 2*e3x))/60

d302:   e2x**2/20

d311:   ( - e2x*e3x)/20

d320:   e3x**2/20
```

differentiation with respect to Qxy(vl):

```
d212:   ( - 5*e2x*e2y*s2 + 3*e2x*e2y - e2x*e3y - e2y*e3x)/30

d221:   ( - e2x*e3y - e2y*e3x + 5*e3x*e3y*s3 - 2*e3x*e3y)/30

d302:   (e2x*e2y)/10

d311:   ( - (e2x*e3y + e2y*e3x))/20

d320:   (e3x*e3y)/10
```

differentiation with respect to Qyy(v1):

d212: (e2y*(- 5*e2y*s2 + 3*e2y - 2*e3y))/60

d221: (e3y*(- 2*e2y + 5*e3y*s3 - 2*e3y))/60

d302: e2y**2/20

d311: (- e2y*e3y)/20

d320: e3y**2/20

differentiation with respect to Qx(v2):

d032: (2*e1x)/5

d041: e1x/5

d122: (5*e1x*s1 - e1x - e3x)/10

d131: (e1x - e3x)/5

d140: (- e3x)/5

d221: (e1x + 5*e3x*s3 - 4*e3x)/10

d230: (- 2*e3x)/5

differentiation with respect to Qy(v2):

d032: (2*e1y)/5

d041: e1y/5

d122: (5*e1y*s1 - e1y - e3y)/10

d131: (e1y - e3y)/5

d140: (- e3y)/5

d221: (e1y + 5*e3y*s3 - 4*e3y)/10

d230: (- 2*e3y)/5

differentiation with respect to Qxx(v2):

```
d032:    e1x**2/20

d122:    (e1x*(5*e1x*s1 - 2*e1x - 2*e3x))/60

d131:    ( - e1x*e3x)/20

d221:    (e3x*( - 2*e1x - 5*e3x*s3 + 3*e3x))/60

d230:    e3x**2/20
```

differentiation with respect to Qxy(v2):

```
d032:    (e1x*e1y)/10

d122:    (5*e1x*e1y*s1 - 2*e1x*e1y - e1x*e3y - e1y*e3x)/30

d131:    ( - (e1x*e3y + e1y*e3x))/20

d221:    ( - e1x*e3y - e1y*e3x - 5*e3x*e3y*s3 + 3*e3x*e3y)/30

d230:    (e3x*e3y)/10
```

differentiation with respect to Qyy(v2):

```
d032:    e1y**2/20

d122:    (e1y*(5*e1y*s1 - 2*e1y - 2*e3y))/60

d131:    ( - e1y*e3y)/20

d221:    (e3y*( - 2*e1y - 5*e3y*s3 + 3*e3y))/60

d230:    e3y**2/20
```

differentiation with respect to Qx(v3):

d014: (- e1x)/5

d023: (- 2*e1x)/5

d104: e2x/5

d113: (- e1x + e2x)/5

d122: (5*e1x*s1 - 4*e1x + e2x)/10

d203: (2*e2x)/5

d212: (- e1x + 5*e2x*s2 - e2x)/10

differentiation with respect to Qy(v3):

d014: (- e1y)/5

d023: (- 2*e1y)/5

d104: e2y/5

d113: (- e1y + e2y)/5

d122: (5*e1y*s1 - 4*e1y + e2y)/10

d203: (2*e2y)/5

d212: (- e1y + 5*e2y*s2 - e2y)/10

differentiation with respect to Qxx(v3):

d023: e1x**2/20

d113: (- e1x*e2x)/20

d122: (e1x*(- 5*e1x*s1 + 3*e1x - 2*e2x))/60

d203: e2x**2/20

d212: (e2x*(- 2*e1x + 5*e2x*s2 - 2*e2x))/60 differentiation with respect to Qxy(v3):

d023: (e1x*e1y)/10

d113: (- (e1x*e2y + e1y*e2x))/20

d122: (- 5*e1x*e1y*s1 + 3*e1x*e1y - e1x*e2y - e1y*e2x)/30

d203: (e2x*e2y)/10

d212: (- e1x*e2y - e1y*e2x + 5*e2x*e2y*s2 - 2*e2x*e2y)/30

differentiation with respect to Qyy(v3):

d023: e1y**2/20

d113: (- e1y*e2y)/20

d122: (e1y*(- 5*e1y*s1 + 3*e1y - 2*e2y))/60

d203: e2y**2/20

d212: (e2y*(- 2*e1y + 5*e2y*s2 - 2*e2y))/60

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 2703	2. GOVT ACCESSION NO. AD-A144663	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MULTIVARIATE SCATTERED DATA DERIVATIVE GENERATION BY FUNCTIONAL MINIMIZATION		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Peter Alfeld		8. CONTRACT OR GRANT NUMBER(s) DAAG29-80-C-0041 DE-AC02-82ER12046 A000
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of Wisconsin 610 Walnut Street Madison, Wisconsin 53706		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 3 - Numerical Analysis and Scientific Computing
11. CONTROLLING OFFICE NAME AND ADDRESS See Item 18 below.		12. REPORT DATE June 1984
		13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

Department of Energy
Washington, D. C. 20545

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Scattered Data
Computer Aided Geometric Design
Bivariate Interpolation
Derivative Generation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Many multivariate interpolation schemes require as data values of derivatives that are not available in a practical application, and that therefore have to be generated suitably. A specific approach to this problem is described that is modeled after univariate spline interpolation. Derivative values are defined by the requirement that a certain functional be minimized

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT - cont'd.

over a suitable space subject to interpolation of given positional data. In principle, the technique can be applied in arbitrarily many variables. The theory is described in general, and particular applications are given in one and two variables. A major tool in the implementation of the technique is the Bézier-Bernstein form of a multivariate polynomial. The technique yields visually pleasing surfaces and is therefore suitable for design applications. It is less suitable for the approximation of derivatives of a given function.

END

FILMED

DTIC